



連載

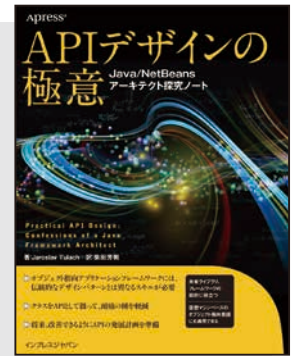
ビブリア・トーク
—私のオススメ—

… 木村功作 (富士通研究所)

API デザインの極意 Java/NetBeans アーキテクト探究ノート

Jaroslav Tulach 著 柴田芳樹 翻訳

(株)インプレスジャパン (2014), 432p., 3,200 円+税, ISBN: 978-4-8443-3591-7



Application Programming Interface (API) は、オペレーティングシステムやミドルウェア等の機能をプログラムから利用するために用いるインタフェースのことであり、かなり昔から存在する概念です。しかし近年でも (といっても随分前からですが) Web API という形で Web 上での機能の提供と利用が活発に行われており、API はいまだにとっても重要な概念であるといえます。情報系の研究者・開発者であれば、今どきのソフトウェアを開発するには必ず何らかの API を利用する必要があったり、研究成果や技術を使ってもらうために自ら API を公開したりするなど、関連が多いのではないかと思います。

本書は、将来の長期にわたる改善・拡張に耐え得るような API を設計するための考え方・ノウハウをまとめた本です。本書は、Java の統合開発環境である NetBeans のアーキテクトを務めた Jaroslav Tulach 氏が著した「Practical API Design: Confessions of a Java Framework Architect」の邦訳であり、翻訳は「Effective Java」等の Java 関連書籍を翻訳・執筆されている柴田芳樹氏によってなされています。

本書以外にも API (特に Web API) の設計に関する良書はいろいろありますが、その中でも私は本書をご一読いただくことをお勧めします。本書の最大の特長は、API の初版設計開始から改版、終息に至るまでのライフサイクルの在りようが、著者自身の成功ばかりとは限らない実体験をもとに語られている点です。約 20 年にわたり多くの開発者を巻き込み、さまざまな拡張プラグインの開発によって API が使い倒され、発展・存続してきた NetBeans のノウハウには、一朝一夕では得られない説得力があります。

本稿では、本書の中でとても印象に残った節タイトルをそのまま使わせていただきつつ、本書の内容をごく一部だけご紹介します。

【この本は、Java だけに役立つのでしょうか】

本書は副題に「Java/NetBeans」を含んでいますが、決して Java 向けに限定された内容の本ではありません。本書では NetBeans の事例を用いてノウハウが語られているため、中には Java の言語仕様や標準クラスライブラリ固有のものが多く出てきます。しかし、本書中の提言や基本的な考え方は特定言語に依存しないものであり、オブジェクト指向に限らずさまざまなプログラミング言語で API を設計する際に参考にすることができます。特に「第一部 理論と正当性」では、優れた API の定義やそれらが必要な理由を哲学やコンピュータの歴史を交えて論じており、非常に有用である上に読みごたえのある内容になっています。

しかしながら、タイトルや表紙デザインの印象、中身の斜め読みだけではそのことが分かりにくく、非常にもったいないと感じています。ぜひとも、Java 以外の言語を愛用する多くの方々にも書店や図書館で本書を手にとってじっくり読んでいただきたいものです。

【最初のバージョンは決して完璧ではない】

これは、本書が述べる API 設計のあるべき姿勢を示唆しています。すなわち、本書では「どれだけみなさんが努力するかに関係なく、リリースされたバージョンはバグを含むでしょうし、ユーザは、あなたの設計が想定している範囲を超えた仕事に使用し

ようとする」という現実を受け入れ、APIは徐々に改善されるものであるという前提で何をすべきかが語られています。

まず、古いバージョンを今まで同様に使えるようにする後方互換性について言及しています。具体的には、「ソース」や「バイナリ」、それらより抽象的な「機能」といった異なるレベルでの後方互換性が存在すること、およびそれらの事例について提示しています。次に、API設計はユースケース指向であるべきと述べています。APIが広く使われるようになるほど、ユーザからAPIに対する要求を直接引き出すことが非常に困難になってきます。そのため、APIを改善する際にはユースケース（ユーザからの使われ方）を想定し、それを実現するような最適化を目指すべき、ということ述べています。

【必要以上に公開しない】

これは、「KISSの原則」等で知られる設計の簡潔さについての普遍的な考え方そのものですが、とても重要なので本書では改めて言及されています。APIを必要以上に公開してしまうことで、拡張が困難になったり想定外の使い方をされたりするような問題が発生することを事例を交えて述べています。

これに関する施策はプログラミング言語によって異なりますが、本書ではJavaについて、

- フィールドよりメソッドを使う
- コンストラクタよりファクトリメソッドを使う
- finalにする
- 不要なsetterを公開しない
- パッケージプライベートにする

- オブジェクトの作成者にのみ特権を与える
 - 深い階層を公開しない
- といったことを推奨しています。

【これからは、無知の時代です】

本書は、APIの提供・利用によるソフトウェア開発の特長を「選択的無知 (selective cluelessness)」という言葉で表現しています。選択的無知とは、開発者は個々のソフトウェア部品の中身のことをまったく知らなくても、それらを組み合わせて大規模なシステムを開発できること、すなわち、開発者は知るべきことと知らなくて良いことを選択することができる、ということを表しています。

本書の著者は、今後ますますソフトウェアが大規模化・複雑化し、開発者に要求される達成度が上がっていく一方で、個々人が理解できるシステムの部分はますます小さくなっていくであろうという予想を立てています。さらに、その予想通りに状況が推移した場合に今後生じるであろうさまざまな問題に対しては、選択的無知が答えを与えてくれると説いています。

これからは、無知の時代になりそうです。皆様在选择的無知でいられるために、本書のAPIに関する知識はぜひ知っておいていただきたいと私は思います。

(2016年2月10日受付)

木村功作 (正会員) kimura.kosaku@jp.fujitsu.com

2010年九州大学大学院システム情報科学府博士後期課程修了。博士(工学)。同年(株)富士通研究所入社。以降、ソフトウェア開発自動化に関する研究開発に従事。